# Contextual and Metadata-based Approach for the Semantic Annotation of Heterogeneous Documents

Mouhamadou Thiam, Nathalie Pernelle, Nacéra Bennacer Seghouani

# Contextual and Metadata-based Approach for the Semantic Annotation of Heterogeneous Documents

Mouhamadou Thiam[1], Nathalie Pernelle[1]
Nacéra Bennacer[2]

[1] LRI, Université Paris-Sud 11, INRIA Futurs, 2-4 rue Jacques Monod, F-91893 Orsay Cedex, France
{Nathalie.Pernelle, Mouhamadou.Thiam}@lri.fr
[2] Supélec, Plateau du Moulon, 91192 Gif-sur-Yvette Cedex, France
{Nacera.Bennacer}@supelec.fr

**Abstract.** In this paper, we present *SHIRI-Annot*, an automatic ontology-driven and unsupervised approach for the semantic annotation of documents which contain more or less structured parts. The aim of this approach is to build an integration system called *SHIRI* [3] which allows the user access to documents related to a specific domain. In this system, the querying process is guided by an ontology of the domain and the answers are only made of the pertinent parts of the documents unlike keywords-based search engines. The ontology describing the domain of interest is defined using a set of concepts, their properties, their relations and the associated cardinalities and it is described using RDFS (Resource Description Framework Schema) language. The *SHIRI-Annot* approach consists of locating and then annotating concept instances and their semantic relations. The locating step combines existing annotation approaches in order to locate instances in the text. The annotation step exploits a set of metadata and a set of logical rule patterns which are automatically instanciated from the domain description. Some of these metadata are provided from the ontology and others are defined specifically for the annotation task. So, the set of logical rules allow annotating parts of the documents with these metadata by taking into account both the semantic relations defined in the ontology and the structural context. The resulting annotations are represented in RDF (Resource Description Framework) language. We show through a preliminary study made on a corpus of HTML documents the usefulness of these specific metadata to represent the heterogeneity of documents. We also illustrate through examples how the *SHIRI* system exploits the metadata to approximate the user queries in order to provide more pertinent instances.

**Keywords:** Semantic Annotation, Metadata, Ontology, HTML, RDF/RDFS, Logical rules.

[3] SHIRI : Digiteo labs project (LRI, SUPELEC)

## 1 Introduction

The Web is a very huge amount of data. The need to automate this data processing, its exploitation by applications and its sharing justify the interest that research carries on the semantic Web. Information available on the Web is mostly in HTML form and thus is more or less syntactically structured. Because of the absence of semantic, the querying of these resources can only be based on keywords. This is not satisfying because it does not ensure answer relevance and the answer is then a whole document. The annotation of web resources with semantic metadata should allow for better interpretations of their contents. Their semantic is defined in a domain description model (an ontology) through the concepts and their relations. Nevertheless, manual annotation is time-consuming. The automation of annotation techniques is a key factor for the future web and its scaling-up. Many works belonging to complementary research fields such as machine learning, knowledge engineering and linguistics investigate the issue of annotation of such documents. Some works are based on supervised approaches or on the existence of structure models in the input documents as in [3], [4], [16], [15] or [7]. But, these works assume some hypotheses which are incompatible with the heterogeneity and the great number of documents. In particular, the existence of a significant and representative number of documents which are manually annotated is an unrealistic hypothesis. Annotation approaches often deal with one kind of structure in the document such as tables in [5] and [9], or text in [2], [1] and [14]. Now, one information may appear in different kinds of structure depending on the document formats. Moreover, one document may contain both structured and unstructured (textual) parts. Each part of one document may describe different instances of different concepts. Except for named entities, instances are often drowned in text, so they are not easily dissociable. Even advanced Natural Language Processing techniques often adapted to very specific corpora could not succeed.

In this paper we present *SHIRI-Annot*, an automatic, ontology-driven and unsupervised annotation approach of HTML documents which contain well structured parts and not well structured ones. The aim of this approach is to build an integration system called *SHIRI*, which allows the user access to documents related to a specific domain. In this system, the querying process is guided by an ontology and answers are only made of pertinent parts of the documents unlike keywords-based search engines. The ontology describing the domain of interest is defined using a set of concepts, their properties, their relations and the associated cardinalities.

In *SHIRI-Annot* approach, an annotation is detached (not embedded) from the content of the document and is associated to its tagged parts called structural units. Each part is annotated as containing one or several instances of different concepts belonging to the ontology of the domain. The approach consists of locating and then annotating concept instances and their semantic relations. In the locating phase, we exploit C-Pankow [2] and Senellart technique [10] which are domain-independant, automatic and unsupervised approaches that locate some concept instances. It concerns concepts whose instances are named entities or

can be delimitated in the text thanks to generic and specific syntactic patterns. In the annotating phase, the purpose is to associate a final annotation to each part of the document using a semantic metadata defined either in the ontology of the domain or in its extension for annotation task. When different instances of different concepts are found in the same structural unit, the specific metadata *PartOfSpeech* is used to annotate this structural unit of the document. This phase also allows inferring the instances of relations linking identified concept instances by exploiting document structures. More precisely, if two instances belong to two imbricated structural units, we assume that there exists a semantic relation linking them. If this relation is identified in the domain ontology, it is instanciated. Otherwise it is the *unamedRelation* specific relation metadata which is instanciated for these two instances. To achieve the annotating phase, we also define a set of logical rule patterns which are automatically generated from the domain description. The resulting annotations are represented using RDF (Resource Description Framework) language. The use of W3C standard RDF/RDFS (RDF Schema) languages allows taking advantage of all technologies around RDF [11] such as the advanced query language SPARQL [12].

We show through a preliminary study made on a corpus of HTML documents the usefulness of these specific metadata in annotating heterogeneous documents belonging to a given domain. We also illustrate through examples how the querying process can exploit these metadata to approximate automatically user queries in order to provide instances which are potentially pertinent.

The paper is organized as follows. In section 2, we present the semantic annotation process defined in *SHIRI-Annot*. In section 3, we show how the contextual metadata are used in the querying process. In the last section, we conclude and give some perspectives.

## 2 Semantic Annotation in *SHIRI-Annot* Approach

*SHIRI* system (figure 1) receives as inputs the domain of interest described in an ontology and a set of documents belonging to this domain. The ontology consists of a set of classes (unary relations) organized in a taxonomy and a set of typed properties (binary relations). These properties can also be organized in a taxonomy of properties. We use the notations $R(C, D)$ to indicate that the domain of the property $R$ is the class $C$ and that its range is $D$ ( $D$ is a class or a literal).

The enrichment process consists first in locating all suitable instances of concepts and then in annotating all document parts. The resulting RDF annotations, the domain ontology and its extension are stored in repository for querying and retrieval. In the following, we detail these components using an example related to call for papers of scientific conferences. This domain is defined formally in figure 2 by a set of concepts, their relations and their cardinalities. The concepts are of type *rdfs:Class* class (represented by a circle in the figure) and the relations are of type *rdf:Property* class (represented by directed arrows from the range to the domain of the relation in the figure). The notation * is

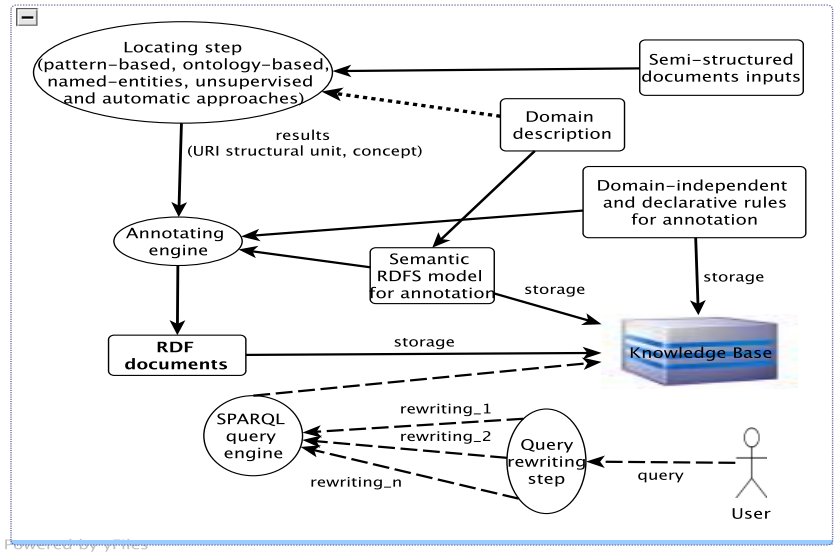


**Fig. 1.** SHIRI Architecture

used to represent cardinalities of relations (by default cardinalities are equal to one). To represent this default cardinality, we extend RDFS by OWL (Ontology Web Language) *functionalProperty* and *inverseFunctionalProperty* constructs.

### 2.1 Locating phase

We first need to exploit the domain ontology and a terminological knowledge in order to locate concept instances. This step is difficult since the vocabulary used to describe text entities, concepts or their property values varies between resources. In the *SHIRI-Annot* locating phase we want to exploit approaches which are domain-independent, automatic and unsupervised. C-Pankow [2] allows us annotating instances of concepts which appear as named entities or as other nominal groups. Using syntactic patterns, instance candidates are extracted from the input documents. Then, using generic Hearst patterns and a search engine such as google, C-Pankow technique proposes for each instance a set of concepts accompanied by a confidence measure for each concept. We choose among proposed concepts those which are similar to one of our domain description concepts and filter them by the provided confidence measure. We also exploit Senellart technique [10] which uses DBLP (Digital Bibliography and Library Project) to identify accurately person names and date instances. Actually, these named entities generally appear in domain descriptions. The figure 3 illustrates the examples of instances we locate by applying C-Pankow [2] and Senellart [10] approaches. If an instance $i$ of a concept $c$ is located in a structural unit *su* a RDF fact *(su, containInstanceOf, c)* is created. The output of the lo-

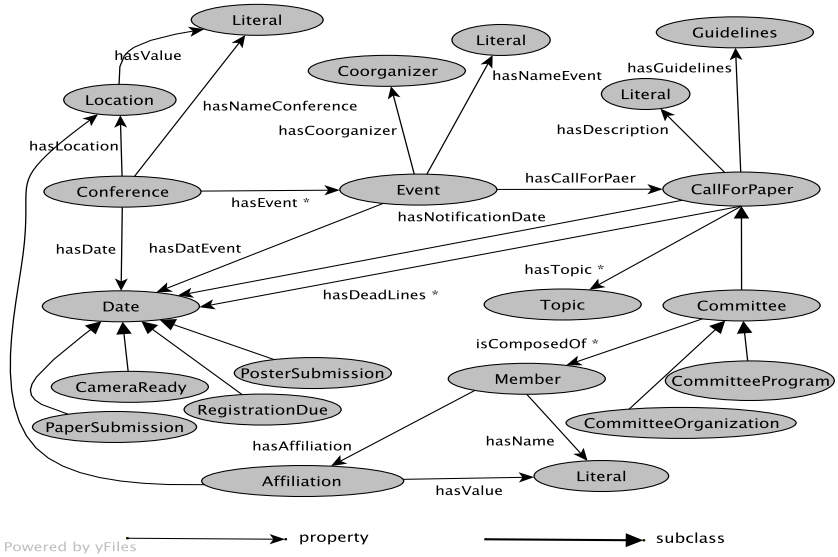


**Fig. 2.** Domain Ontology

cating phase is a set of RDF triplets with the property *containInstanceOf* whose domain is the *StructuralUnit* class and its range is the *Concept* class. If one instance is annotated differently by these approaches, an heuristic combining the annotation measures of the approaches can be applied to determine the appropriate annotation. Otherwise, the following annotation rules take into account this case.

### 2.2 Annotation phase

The purpose of the annotation phase is to associate a final annotation to each part of the document using a semantic metadata defined either in the domain ontology or in its extension. This phase also allows inferring the instances of relations relating identified concept instances by exploiting document structures. More precisely, if two instances belong to two imbricated structural units, we infer that they instanciate a semantic relation belonging to the domain description model. Indeed, we assume that there exists a semantic link between instances found in imbricated parts of the documents. To achieve this, we define a set of specific metadata required by the annotation task and a set of Horn first order logic (FOL) rule patterns which are automatically instanciated from the domain description.

**Domain Ontology Extension for Annotation** The ontology describing the domain is automatically extended with specific metadata defined to annotate

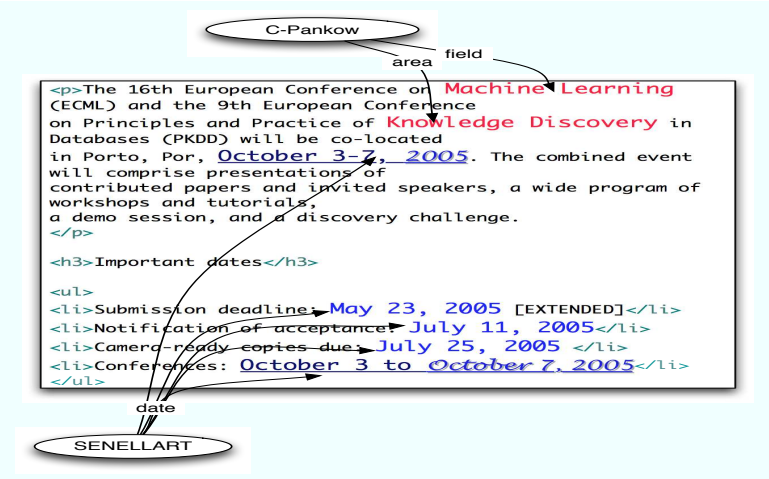


**Fig. 3.** Examples of nstances obtained in locating phase

parts of the document which are not well-structured and to annotate potential semantic links (see figure 4).

First, the RDF class *Concept* which represents a superclass for all concepts belonging to the ontology is added as a subclass of the *Metadata* class.

Then, we add the RDF classes *PartOfSpeech* as a subclass of the class *Metadata*. This class is defined in order to annotate structural units which contain instances of different metadata and in particular those that are not easily dissociable. Their types are then kept using *isIndexedBy* RDF property. Thus, the *isIndexedBy* property values constitute a sort of indexation for this part of speech. The dissociable instances that are contained in a *PartOfSpeech* instance are annotated using the RDF property *containMetadataInstance*. Moreover, the instances of the *Metadata* class can contain instances of the *PartOfSpeech* class via *containPartOfSpeech* RDF property. For example, we assume that in the following part of document " *ECAI 2008, the 18th conference in this series, is jointly organized by the European coordinating Commitee on Artificial Intelligence the university of Patras and the Hellenic Artificial Intelligence Society* " the instances of *Date, Conference, Topic, Location* have been found in the locating phase. This structural unit is annotated by *PartOfSpeech* metadata and indexed by the concept names.

We also define the RDF classes *SetOfX* as subclasses of the class *Metadata*. The purpose is in this case to annotate the part of document containing a set of instances of the same concept without separating them. For each concept $X$ which is the range of a not functional property $P$, we add a RDFS class *SetOfX* as subclass of *Metadata* and a RDF property *hasSetOfX* to relate it to the concept $X$. Thus, the property $P$ links *SetOfX* class to $X$ class. For example, since the property *hasTopic* is not functional and its range is *Topic*, we add *SetOfTopic* class and *hasSetOfTopic* property as described in figure 4.

The initial domain of *hasTopic* (*CallForPaper* in the figure 4) is also domain of *hasSetOfTopic*. The range of *hasSetOfTopic* is *SetOfTopic*. We apply a similar reasoning for not inverse functional properties. For example, the text of the ESWC call for paper contains " *searching, querying, visualizing, navigating and browsing the semantic web* " where topics should be separated in "*searching in the semantic web* " and "*querying the semantic web* ". We annotate this part by the metadata *SetOfTopic* and the relation *hasSetOfTopic* can be instanciated to link it to *CallForPaper* instance of the conference.

To keep links between concept instances, we use either the semantic relations defined in the domain ontology or the RDF property *unnamedRelation* property for unidentified ones. This new property subsumes all the domain description properties.

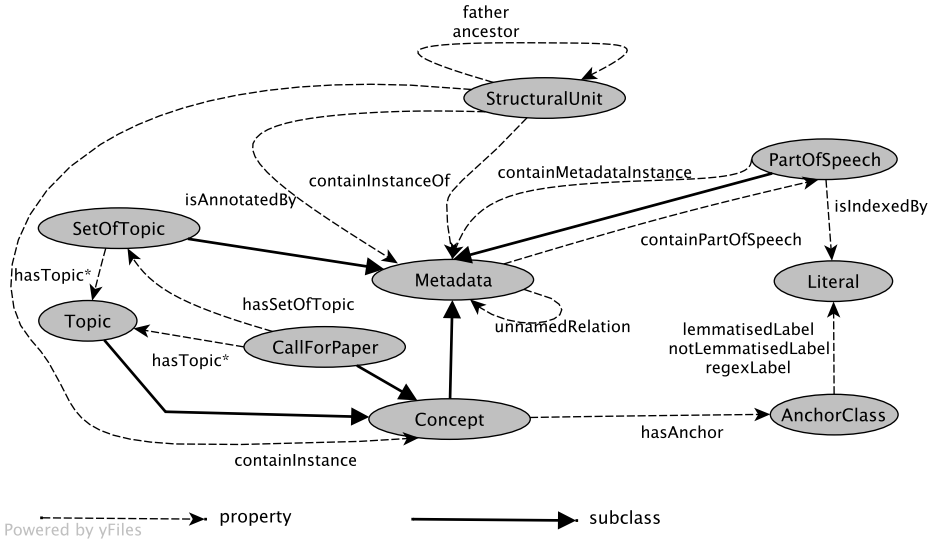


**Fig. 4.** Domain Ontology Extension for Annotation

**Annotation Rules** The logical rules are generated and applied to the output of the locating step (*containInstanceOf* RDF triplets) into a two step process. In the initialization step, the purpose is to instanciate the *Concept*, *PartOfSpeech* and *SetOfX* metadata. Let $x$, $y$ be node variables, we note father(x,y)[4] and ancestor(x,y) the fact that the node $x$ is father, respectively ancestor of $y$ in the HTML tree structure.

[4] *property(x, y)* represents the triple *(x, property, y)*

- If a node $x$ does not contain any concept instance, it is annotated using *PartOfSpeech* metadata : we add the fact *PartOfSpeech(x)* [5].
- If a node $x$ contains exactly one instance of a concept $c$, this node is annotated by the metadata $c$ : we add the fact $c(x)$. For all concepts that can be multivalued in a relation, the node is annotated by the corresponding class *setOfX* in the initialization step. For instance, the metadata *setOfTopic* is used instead of the concept *topic* which is multivalued in the relation *hasTopic*.
- If a node $x$ contains instances of different concepts, this node is annotated using *PartOfSpeech* metadata : we add the fact *partOfSpeech(x)*. This node is related via *isIndexedBy* property to each concept name "$c_i$" of identified instances. For all $c_i$, we add: *isIndexedBy(x, "$c_i$")*.
- If two nodes $x$, $y$ such that $father(x, y)$ are annotated by the same concept $c$, the annotation associated to y is eliminated. We consider that these two instances refer to the same one and thus the node $x$ delimitates more precisely this instance.

The aim of the next step is to instanciate the semantic relations existing between instances and to propagate the *isIndexedBy* property of *partOfSpeech(x)* instances to ancestor nodes. Let $C$ be the set of concepts and let $R$ be the set of relations in the domain ontology.

- $\forall$ $c_i \in C$ , we add the following rule :
  $c_i(x) \wedge PartOfSpeech(y) \wedge father(y, x) \Rightarrow$
  $containMetadataInstance(y, x) \wedge isIndexedBy(y, "c_i")$
  where "$c_i$" is the concept name literal.
  This rule expresses that if a *PartOfSpeech* node is the father of a concept node, the relation *containMetadataInstance* is instantiated in order to relate these two nodes and the *PartOfSpeech* node is indexed by the concept.
- $PartOfSpeech(x) \wedge father(y, x) \Rightarrow containPartOfSpeech(y, x)$
  This rule expresses that if a node is the father of a *PartOfSpeech* node, the father node is related to the *PartOfSpeech* node by a *containPartOfSpeech*.
- $PartOfSpeech(x) \wedge ancestor(y, x) \wedge PartOfSpeech(y) \wedge isIndexedBy(x, index) \Rightarrow isIndexedBy(y, index)$
  This inheritance rule expresses that *PartOfSpeech* nodes are indexed by all metadata located in their descendant nodes.
- $\forall$ $r(c, d) \in R \cup \{\text{unnamedRelation}\}$, and $\forall$ $c_i$ and $c_j$ such that $c$ subsumes $c_i$ and $d$ subsumes $c_j$
  $c_i(x) \wedge c_j(y) \wedge father(x, y) \Rightarrow r(x, y)$
  $c_i(x) \wedge c_j(y) \wedge father(y, x) \Rightarrow r(x, y)$
  This means that, for all pairs of nodes $x$, $y$ annotated respectively by $c_i$ and $c_j$ concepts, if there exists a semantic relation $r$ between $c_i$ and $c_j$ in the RDFS model, we add $r(x, y)$. If no relation of the initial RDFS model is found, the nodes are only related via the *unnamedRelation* relation.

[5] *metadata(x)* represents the triple *(x, isAnnotatedBy, metadata)*

In the following, we call an empty node, each node which is annotated by *PartOfSpeech* metadata and which is not related to any concept name via *isIndexedBy* relation. After applying the set of annotation rules, all descendant nodes of an empty node are also empty nodes. We are interested in empty nodes having as parent a node which is annotated by the metadata *SetOfX* (for example *setOfTopic*). In this case, if the HTML tags used for child nodes are similar and show a repetitive HTML structure which is the case of lists (ul or ol) or a table we infer that each structural unit contains an instance of the metadata *Concept* (for example *Topic*). The example bellow illustrates two cases. In the first case the different topics appear in different structural units with the same HTML tag. So, the annotation can associate each structural unit to the metadata *Topic*. In the second case, all the topics are described in the same structural unit. So only the metadata *SetOfTopic* is used in the annotation. The figure 5 illustrates the locating and annotating phases. In the left part, after the locating phase, each structural unit is identified as containing a set of concepts instances. The right part of the figure shows the set of instantiations obtained after applying the annotation rules. In the worst case, if no structural unit is identified as a metadata instance, all nodes are empty except the root node which is annotated by the concept *Conference*. In this case where the document is not well structured and metadata instances are not dissociable, structural units are mostly annotated by *PartOfSpeech* metadata and each one collects all the concepts whose instances were located in. In the best case, structural units are mostly annotated by *Concept* metadata and related by named semantic relations.

| | |
|---|---|
| ...the topics of interest : <ul><br><li>Computer architectures for public-key</li><br><li>Set-key cryptosystems</li><br><li>Reconfigurable computing</li><br></ul>  | <br>... Topics of interest include (but are not limited to):<br>Authentication, Case studies, Access control, Cryptographic algorithms, Accounting and auditing, Cryptographic protocols. Thus are some of topics but other areas related to the domain of interest are welcome.<br> |

**Preliminary Experiments** The corpus is composed of 444 HTML documents coming from 33 web sites about scientific conferences in computer science. All web pages are processed using a cleaning engine based on HtmlCleaner [13] to obtain well-formed ones.

The aim of this preliminary study is to show the usefulness of the specific metadata to represent the heterogeneity of documents. We have focused on one metadata *Topic* which is multivalued for a *CallForPaper*. Furthermore these metadata instances appear in different structuration forms in the documents.

By inventorying all structural units containing *Topic* concept instances, we could show that these instances appear in structural units that are more or less struc-

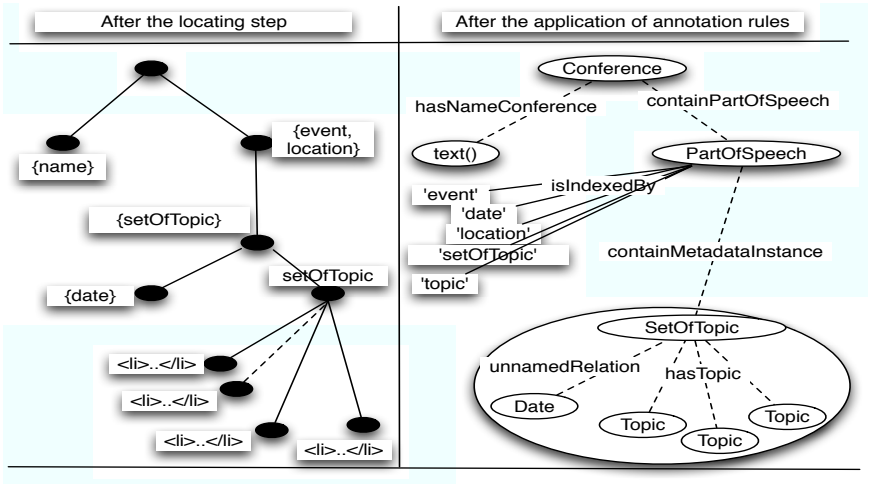


**Fig. 5.** A tree example after locating and annotating

tured. More precisely, about 54% of its instances appear in distinct structural units of a document. About 23,2% of its instances appear in the same structural unit of a document. Finally, about 22,8% of these instances are encapsulated in structural units which contain other instances of different concepts (case corresponding to *PartOfSpeech* metadata). So we can observe that instances of metadata for multivalued properties appear with numerous frequencies in the three levels of structuring.
In our results, we consider, for each conference, either topic's instances are exactly identified denoted by c or partially discovered (case where some instances are forgotten), or identified but mixed with other instances of other metadata or wrong. The annotation is evaluated using precision and recall measures. The recall, expressed as a percentage, measures the overall proportion of relevant instances found by the system and the precision measures the percentage of relevant instances among those annotated. We obtain a recall of 65,1% and a precision of 84,3%. The recall will be of course higher when annotation rules concerning the *PartOfSpeech* metadata will be implemented since they represent 23,8% of cases.

## 3 Annotated Documents Querying

To query resulting RDF documents, we plan to use SPARQL ([12]). The querying interface will be based on concepts and relations of the RDFS model to help users to formulate their queries in SPARQL. Hence, queries can be automatically rewritten using all possible paths of the extended semantic model in order to exploit the annotations associated to less structured parts of documents. Let us suppose a user who queries the names of conferences which have *cryptography* among their topics.

*Case 1:* The document contains an instance of *event* that contains an instance of *setOfTopic* which contains an instance of *topic* which contains the word *cryptograph*. The query has the following rewriting:

```
PREFIX cfp: <http://corpusCFP/thiam/ontology#>
SELECT ?name
WHERE { ?c cfp:hasNameConference ?name .
 ?c cfp:hasEvent ?e .
 ?e cfp:hasSetOfTopic ?st .
 ?st cfp:hasTopic ?t .
 FILTER regex(?t, "cryptograph", "i")
}
```

*Case 2:* The document contains an instance of *event* which contains an instance of *setOfTopic* which contains the word *cryptograph* in its textual value. The query is rewritten as follow:

```
PREFIX cfp: <http://corpusCFP/thiam/ontology#>
SELECT ?name
WHERE { ?c cfp:hasNameConference ?name .
 ?c cfp:hasEvent ?e .
 ?e cfp:hasSetOfTopic ?st .
 FILTER regex(?st, "cryptograph", "i")
}
```

*Case 3 :* The document contains an instance of *PartOfSpeech* which is indexed by the textual value *"topic"*. The query has the following rewriting:

```
PREFIX cfp: <http://corpusCFP/thiam/ontology#>
SELECT ?name
WHERE { ?c cfp:hasNameConference ?name .
 ?c cfp:ancestor ?pos .
 ?pos cfp:isIndexedBy ?m .
 FILTER (regex(?m, "topic", "i") && regex(?pos, "cryptograph", "i"))
}
```

The *SHIRI-Annot* annotation approach provides a basis for ranking answers. The querying system can suggest all possible instances by setting a relevance weighting measure for each one according to the semantic annotations associated to the structural units in which these instances are located. Indeed, An instance *Event* found in a structural unit annotated by *Event* concept and related to a *Conference* structural unit by *hasEvent* property is more relevant than an instance *Event* found in structural unit annotated by *partOfSpeech* metadata or related to a *Conference* structural unit by *unnamedRelation* property. A similar reasoning can be done for instances found by exploiting subsumption relations, the system can suggest by relaxing a query an instance of *Commitee* instead of an instance of *CommiteeProgram*.

## 4 Conclusion and perspectives

In this paper, we present an automatic and unsupervised approach for semantic annotation of heterogeneous HTML documents based on a description of the domain of interest. The existing approaches are generally adapted to deal with either textual documents or structured documents. Nevertheless, one HTML document is often heterogeneously structured since it contains both well-structured parts (tables, lists, ...) and unstructured textual parts. *SHIRI-Annot* is an approach which combines different kinds of annotation and indexation methods. To find the instances of concepts and relations in each part of the documents, we define a set of Horn FOL annotation rules that take into account both the semantic relations of the domain model and the heterogeneity of document structures. Moreover, these rules allow the instantiation of relations between identified instances located in related parts. In our approach we take advantage of RDF/RDFS expressivity and flexibility to represent the domain and the resulting annotated documents. In particular, the fact that the annotations are not embedded in the documents allows associating metadata related to different domains to one located instance.
We also show how the extended semantic model can be used to formulate and to approximate queries in order to adapt them to the various levels of precision of the annotation. In this way, the querying system answers as precisely as possible to user queries and provides a relevant measure for each answer.
Besides, the annotated parts of documents can be gathered to populate the ontology of the considered domain. The more *SHIRI-Annot* system is used the more it is efficient.
The obtained results are encouraging and we will go on implementing all annotation rules. Moreover, we plan to exploit finer Natural Processing Language techniques to improve the locating step but also to automatically (or semi-automatically) complete the set of concepts and relations.
We also plan to apply our approach to other domains like e-commerce web sites.

## References


1. Alani H., Kim S., Millard D.E., Weal M.J., Hall W., Lewis P.H., Shadbolt N.: Using Protégé for Automatic Ontology Instantiation. In Proceeding of 7th International Protégé Conference, 2004.
2. Cimiano P., Handschuh S., Staab S.: Gimme'The Context : Context Driven Automatic Semantic Annotation With C-PANKOW. WWW conference, 2005.
3. Crescenzi V., Mecca G., Merialdo P.: RoadRunner : Towards Automatic Data Extraction from Large Web Sites. Very Large Data Bases Conference (VLDB), 2001.
4. Davulcu H., Vadrevu S. and Nagarajan S.: OntoMiner : Automated Metadata and instance Mining from News Websites. The International Journal of Web and Grid Services (IJWGS), Vol. 1, No. 2, pp. 196-221, Inderscience Publishers, 2005.
5. Gagliardi H., Haemmerlé O., Pernelle N., Sais F.: An automatic ontology-based approach to enrich tables semantically. Proceedings of AAAI, The first International Workshop on Context and Ontologies : Theory, Practice and Applications, 2005.

6. Borislav P., Atanas K., Angel K., Dimitar M., Damyan O., Miroslav G.: KIM - Semantic Annotation Platform. Journal of Natural Language Engineering vol 10 issue 3-4, Cambridge University Press, pages 375-392, 2004.
7. Baumgartner R. and Flesca S. and Gottlob G: Visual Web Information Extraction with Lixto. The VLDB Journal, pages 119-128, 2001, citeseer.ist.psu.edu/baumgartner01visual.html
8. Jose Kahan, Marja-Riitta Koivunen, Eric Prud'Hommeaux, and Ralph R. Swick: Annotea: An Open RDF Infrastructure for Shared Web Annotations. in Proc. of the WWW10 International Conference, 2001
9. Gilleron R. and Marty P. and Tommasi M. and Torre F: Interactive Tuples Extraction from Semi-Structured Data. in IEEE / WIC / ACM International Conference on Web Intelligence, 2006.
10. Senellart P. : Understanding the Hidden Web. PHD Thesis, University of Paris 11, December 2007.
11. http://www.w3.org/rdf.
12. http://www.w3.org/TR/rdf-sparql-query.
13. http://htmlcleaner.sourceforge.net/
14. Dingli, A., Ciravegna, F. and Wilks, Y., Automatic Semantic Annotation using Unsupervised Information Extraction and Integration in K-CAP 2003 Workshop on Knowledge Markup and Semantic Annotation, (2003).
15. Handschuh, S., Staab, S. and Ciravogna, F., S CREAM Semi automatic CREAtion of Metadata in SAAKM 2002 Semantic Authoring, Annotation Knowledge Markup Preliminary Workshop Programme, (2002).
16. Soderland S. Learning information extraction rules for semi-structured and free text. Machine Learning, 34(1-3):233-272, 1999.
17. Bikel D. M., Miller S., Schwartz R., Weischedel, R. M. Nymble : A high-performance learning name-finder. In Proceedings of the 5th Conference on Applied Natural Language Processing, pp 194-201,1997.
18. Freitag D. Information Extraction from HTML : Application of a General Machine Learning Approach. In AAAI/IAAI proceedings, 517-523, 1998.